Graphical Methods for Defense Against False-data Injection Attacks on Power System State Estimation

Suzhi Bi, *Student Member, IEEE* and Ying Jun (Angela) Zhang, *Senior Member, IEEE*

***Abstract*—The normal operation of power system relies on accurate state estimation that faithfully reflects the physical aspects of the electrical power grids. However, recent research shows that carefully synthesized false-data injection attacks can bypass the security system and introduce arbitrary errors to state estimates. In this paper, we use graphical methods to study defending mechanisms against false-data injection attacks on power system state estimation. By securing carefully selected meter measurements, no false data injection attack can be launched to compromise any set of state estimates. We characterize the optimal protection problem, which protects the state estimates with minimum number of measurements, as a variant Steiner tree problem in a graph. Based on the graphical characterization, we propose both exact and reduced-complexity approximation algorithms. In particular, we show that the proposed tree-pruning based approximation algorithm significantly reduces computational complexity, while yielding negligible performance degradation compared with the optimal algorithms. The advantageous performance of the proposed defending mechanisms is verified in IEEE standard power system testcases.**

***Index Terms*—False-data injection attack, power system state estimation, smart grid security, graph algorithms.**

I. Introduction

A. Motivations and summery of contributions

THE current power systems are continuously monitored and controlled by EMS/SCADA (Energy Management System and Supervisory Control and Data Acquisition) systems in order to maintain the operating conditions in a normal and secure state [1]. In particular, the SCADA host at the control center processes the received meter measurements using a state estimator, which filters the incorrect data and derives the optimal estimate of the system states. These state estimates will then be passed on to all the EMS application functions, such as optimal power flow, etc, to control the physical aspects of the electrical power grids.

However, the integrity of state estimation is under mounting threat as we gradually transform the current electricity infrastructures to future smart power grids. Smart power grids are more open to and physically accessible by the outside networks, such as office local area networks and smart meters that allow two-way communications between energy consumers and suppliers. With these entry points introduced to the power system, potential complex and collaborating malicious attacks are brought in as well. Liu *et al*. [2] showed that a new false-data injection attack could circumvent bad data detection (BDD) in today's SCADA system and introduce arbitrary errors to state estimates without being detected. Such an attack is referred to as an undetectable false-data injection attack. A recent experiment in [3] demonstrates that the attack can cause a state-of-the-art EMS/SCADA state estimator to produce a bias of more than 50% of the nominal value without triggering the BDD alarm. Biased estimates could directly lead to serious social and economical consequences. For instance, [4] and [5] showed that attackers equipped with data injection can manipulate the electricity price in power market. Worse still, [6] warned that the attack can even cause regional blackout.

A common approach to mitigate false-data injection attack is to secure meter measurements by, for example, guards, video monitoring, or tamper-proof communication systems, to evade malicious injections [7]–[9]. Recent studies have proposed a number of methods to select meter measurements for protection. For instance, [7] proved that it is necessary and sufficient to protect a set of *basic measurements* so that no undetectable false-data injection attack can be launched. However, the protection scheme in [7] is costly in that the size of a set of *basic measurements* is the same as the number of unknown state variables in the state estimation problem, which could be up to several hundred in a large-scale power system. On the other hand, despite the vast size of unknown state estimates, only part of the them are indeed considered to be critical to maintain the normal operations of power systems, such as those critical state variables used to maintain voltage stability and the synchronism among generators [10], [11]. Therefore, it is valuable to devise a method that gives priority to defending those state estimates that serve our best interests.

In this paper, we focus on using graphical methods to derive efficient strategies that defend any set of critical state estimates with minimum number of secure measurements. Our detailed contributions are listed as follows,

- We derive conditions to select a set of meter measurements, so that no undetectable attack can be launched to compromise the critical state estimates if the selected meters are secured. The conditions are particularly useful in formulating the optimal protection problem that defends the critical states with a minimum cost.
- We characterize the optimal protection problem as a variant Steiner tree problem in a graph. Then, two exact solution methods are proposed, including a Steiner vertex enumeration algorithm and a mixed integer linear programming (MILP) formulation derived from a network

This work was supported in part by the National Natural Science Foundation of China (Project number 61201261), the National Basic Research Program (973 program Program number 61101132) and the Competitive Earmarked Research Grant (Project Number 419509) established under the University Grant Committee of Hong Kong.

S. Bi and Y. J. Zhang are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (Email: {bsz009, yjzhang}@ie.cuhk.edu.hk)

flow model. In particular, the proposed MILP formulation reduces the computational complexity by exploiting the graphical structure of the optimal solution.
- To tackle the intractability of the problem, we also propose a polynomial-time tree-pruning heuristic (TPH) algorithm. With a proper parameter, simulation results show that it yields close-to-optimal solution, while significantly reducing the computational complexity. For instance, the TPH solves a problem of a 300-bus testcase in seconds, which may take days by the MILP formulation.

B. Related works

State estimate protection is closely related to the concept of power network observability. The conventional power network observability analysis studies whether a unique estimate of all unknown state variables can be determined from the measurements [1]. From the attacker's perspective, [2] proved that an undetectable attack can be formulated if removing the measurements it compromises will make the power system unobservable. Conversely, [7] showed that no undetectable attack can be formulated if the power system is observable from the protected meter measurements. In this paper, we extend the conventional wisdom of power network observability to a generalized *state estimate observability* to study the protection mechanisms for any set of critical state estimates.

Graphical method is commonly used for power system observability analysis. The early work by Krumpholz *et al*. [12] stated that a power system is observable if and only if it contains a spanning tree that satisfies certain measurement-to-transmission-line mapping rules. A follow-up work presented a max-flow method to find such mapping to examine the observability of a power network [13]. Few recent papers also applied graphical methods to study the attack/defending mechanisms of false-data injection. For instance, based on the results in [12], [14] proposed an algorithm to quantify the minimum-effort undetectable attack, i.e. the non-trivial attack that compromises least number of meters without being detected. Besides, [15] used a min-cut relaxation method to calculate the security indices defined in [16] to quantify the resistance of meter measurements in the presence of injection attack. Similar min-cut approach was also applied in [17] to identify the critical points in the measurement set, the loss of which would render the power system unobservable.

The problem of defending a set of critical state estimates against undetectable attack was first studied in our earlier work [18], where we proposed an arithmetic greedy algorithm which finds the minimum set of protected meter measurements by gradually expanding the set of secure state estimates. However, the computational complexity of the greedy algorithm can be prohibitively high in large scale power systems. For instance, it may take years to obtain a solution in a 57-bus system. In contrast, we study in this paper the optimal protection from a graphical perspective. By exploiting the graphical structures of the optimal solution, the proposed MILP formulation obtains the optimal solution with significantly reduced complexity. In addition, we also propose a pruning-based heuristic that yields near-optimal solutions in polynomial time.

The rest of this paper is organized as follows. In Section II, we introduce some preliminaries about state estimation and false-data injection attack. We characterize the optimal protection problem in a graph in Section III and propose efficient algorithms in Section IV. Simulation results are presented in Section V. Finally, the paper is concluded in Section VI.

II. Preliminary

A. DC measurement model and state estimation

We consider the linearized power network state estimation problem in a steady-state power system with $n+1$ buses. The states of the power system include the bus voltage phase angles and voltage magnitudes. The voltage magnitudes can often be directly measured, while the values of phase angles need to be obtained from state estimation [19]. In the linearized (DC) measurement model, we assume the knowledge of voltage magnitudes at all buses and estimate the phase angles based on the active power measurements, i.e. the active power flows along the power lines and active power injections at buses [1]. By choosing an arbitrary bus as the reference with zero phase angle, the network state consisting of the n unknown voltage phase angles is captured in a vector $\boldsymbol{\theta} = (\theta_1, \theta_2, .., \theta_n)'$. In the DC measurement model, the m received measurements $\mathbf{z} = (z_1, z_2, .., z_m)'$ are related to the network states as

$$\mathbf{z} = \mathbf{H}\boldsymbol{\theta} + \mathbf{e}. \tag{1}$$

Here, $\mathbf{H}$ is the measurement Jacobian matrix [1]. $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is independent measurement noise with covariance $\mathbf{R}$. When $\mathbf{H}$ is full column rank, i.e. $rank(\mathbf{H}) = n$, the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ is given by

$$\hat{\boldsymbol{\theta}} = \left(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z} \triangleq \mathbf{P}\mathbf{z}. \tag{2}$$

Since $rank(\mathbf{H}) \leq m$, i.e. the number of rows in $\mathbf{H}$, at least n meters are needed to derive a unique state estimation. Meanwhile, the other $m-n$ measurements provide the redundancy to improve the resistance against random errors.

Errors could be introduced due to various reasons, such as device misconfiguration and malicious attacks. The current power systems use BDD mechanism to remove the bad data assuming that the errors are random and unstructured. It calculates the residual $\mathbf{r} = \mathbf{z} - \mathbf{H}\hat{\boldsymbol{\theta}}$ and compares its l_2-norm with a prescribed threshold τ. A measurement $\mathbf{z}$ is identified as a bad data measurement if

$$r = ||\mathbf{z} - \mathbf{H}\hat{\boldsymbol{\theta}}|| = ||(\mathbf{I} - \mathbf{H}\mathbf{P})\mathbf{e}|| > \tau. \tag{3}$$

Otherwise, $\mathbf{z}$ is considered as a normal measurement.

B. Undetectable attacks and protection model

Suppose that attackers inject malicious data $\mathbf{a} = (a_1, a_2, .., a_m)'$ into measurements. Then, the received measurements become

$$\tilde{\mathbf{z}} = \mathbf{H}\hat{\boldsymbol{\theta}} + \mathbf{e} + \mathbf{a}. \tag{4}$$

In general, $\mathbf{a}$ is likely to be identified by the BDD if it is unstructured. Nevertheless, it is found in [2] that some well-structured injections, such as those with $\mathbf{a} = \mathbf{H}\mathbf{c}$, can bypass

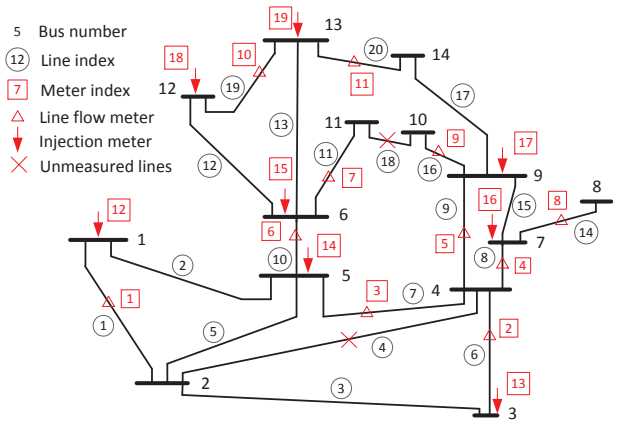

Fig. 1. A measurement placement for the IEEE 14-bus testcase.

BDD. Here $\mathbf{c} = (c_1, c_2, .., c_n)'$ is a random vector. This can be verified by calculating the residual in (4), where

$$\tilde{r} = ||\tilde{\mathbf{z}} - \mathbf{H}\mathbf{P}\tilde{\mathbf{z}}|| = ||\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\boldsymbol{\theta}} + \mathbf{c})|| = ||\mathbf{z} - \mathbf{H}\hat{\boldsymbol{\theta}}||. \quad (5)$$

The same residual is obtained as if no malicious data were injected. Therefore, a structured attack $\mathbf{a} = \mathbf{H}\mathbf{c}$ will not be detected by BDD. In this case, the system operator would mistake $\hat{\boldsymbol{\theta}} + \mathbf{c}$ for a valid estimate, and thus an error vector $\mathbf{c}$ has been introduced without being detected.

The risks of undetectable attacks can be mitigated if the system operator can secure measurements to evade malicious injections. Within this context, we assume that the system operator's objective is to ensure that no undetectable attack can be formulated to compromise a given set of state estimates $\mathcal{D} \subseteq \mathcal{I}$, where $\mathcal{I}$ is the set of all unknown state estimates. That is, $c_i = 0$ for all $i \in \mathcal{D}$. This is achieved by securing a set of meter measurements $\mathcal{P} \subseteq \mathcal{M}$, where $\mathcal{M}$ is the set of all the meters. In other words, attackers are not able to inject false data to any protected meter measurement, i.e. $a_i = 0, \forall i \in \mathcal{P}$.

From [18], securing a set of meters $\mathcal{P}$ would eliminate the possibility of undetectable attack to compromise a set of state variables $\mathcal{D}$, if and only if

$$rank\left(\mathbf{H}_{\{\mathcal{P}\},*}\right) = rank\left(\mathbf{H}_{\{\mathcal{P}\},\{\mathcal{I}\setminus\mathcal{D}\}}\right) + |\mathcal{D}|. \quad (6)$$

Here, $\mathbf{H}_{\{\mathcal{P}\},*}$ is the submatrix of $\mathbf{H}$ including the rows that correspond to $\mathcal{P}$ and $\mathbf{H}_{\{\mathcal{P}\},\{\mathcal{I}\setminus\mathcal{D}\}}$ is the submatrix of $\mathbf{H}_{\{\mathcal{P}\},*}$ excluding the columns that correspond to $\mathcal{D}$. Naturally, we are interested in minimizing the cost to protect the critical states $\mathcal{D}$. For simplicity, we assume a fixed cost, e.g. manpower or surveillance installation cost, of securing each meter for the time being. This requires solving the following problem

$$\begin{aligned} \underset{\mathcal{P}\subseteq\mathcal{M}}{\text{minimize}} \quad & |\mathcal{P}| \\ \text{subject to} \quad & rank\left(\mathbf{H}_{\{\mathcal{P}\},*}\right) = rank\left(\mathbf{H}_{\{\mathcal{P}\},\{\mathcal{I}\setminus\mathcal{D}\}}\right) + |\mathcal{D}|, \end{aligned} \quad (7)$$

which is proved to be an *NP-hard* problem in the next section.

III. Graphical Characterizations of Optimal State Estimate Protection

Interestingly, we show that (7) can be characterized as a variant Steiner tree problem in a graph. The results will be used in the next section to develop efficient graphical algorithms.

A. Network observability and state estimate protection

A power network can be described in an undirected graph, where vertices and edges represent buses and transmission lines, respectively. We use $e_i^{(h)}$ and $e_i^{(t)}$ to denote the two vertices connected to the edge e_i, and $\mathcal{N}_j$ to denote the set of edges incident to vertex v_j. First, we have the following definitions on meter measurements.

Definition 1: The flow meter on transmission line e_i *measures* the edge e_i, and the two vertices $e_i^{(h)}$ and $e_i^{(t)}$. The injection meter at bus v_j *measures* the edge set $\{e_i \mid e_i \in \mathcal{N}_j\}$, and vertex set $\left\{e_i^{(h)}, e_i^{(t)} \mid e_i \in \mathcal{N}_j\right\}$.

Definition 2: For a set of meters $\bar{\mathcal{M}} \subseteq \mathcal{M}$, $\bar{G}\left(\bar{\mathcal{M}}\right) = \left(\bar{\mathcal{V}}, \bar{\mathcal{E}}\right)$ is the *measured subnetwork of* $\bar{\mathcal{M}}$, including all the vertices $\bar{\mathcal{V}}$ and edges $\bar{\mathcal{E}}$ measured by $\bar{\mathcal{M}}$. In particular, $\bar{G}\left(\mathcal{M}\right)$ is referred to as the *measured full network*.

A measurement placement of a 14-bus testcase is presented in Fig. 1. For instance, the flow meter r_6 measures edge e_{10} and vertices v_5 and v_6. The injection meter r_{12} measures edges e_1 and e_2, and vertices v_1, v_2 and v_5. The measured subnetwork of $\bar{\mathcal{M}} = \{r_6, r_{12}\}$ is a closed network consisting of vertices $\bar{\mathcal{V}} = \{v_1, v_2, v_5, v_6\}$ and edges $\bar{\mathcal{E}} = \{e_1, e_2, e_{10}\}$.

The conventional power network observability analysis studies whether a unique estimate of all unknown state variables can be determined [1]. Here, we extend the concept of network observability to a generalized state estimate observability. With a bit abuse of notation, we use a set of vertices $\mathcal{V}'$ to denote the corresponding state variables.

Definition 3: A set of state variables $\mathcal{D} \subseteq \mathcal{I}$ is *observable* from a set of meters $\mathcal{P} \subseteq \mathcal{M}$, if and only if a unique estimate of $\mathcal{D}$ can be obtained from the measurements $\mathcal{P}$. Besides, $\mathcal{P}$ is a *basic measurement set* of $\mathcal{D}$, if $\mathcal{D}$ is observable from $\mathcal{P}$ and $|\mathcal{P}| = |\mathcal{D}|$.

Remark 1: The conventional definition of network observability is a special case with $\mathcal{D} = \mathcal{I}$ and $\mathcal{P} = \mathcal{M}$. A basic measurement set of $\mathcal{D}$ is the minimum measurement set that ensures the observability of $\mathcal{D}$. However, not all $\mathcal{D}$'s have a basic measurement set.

Definition 4: A measured subnetwork $\bar{G}\left(\mathcal{P}\right) = \left(\mathcal{V}', \mathcal{E}'\right)$ is an *observable subnetwork* if and only if all the unknown state variables $\mathcal{S}$ in the subnetwork is observable from $\mathcal{P}$, i.e.

$$rank\left(\mathbf{H}_{\{\mathcal{P}\},\{\mathcal{S}\}}\right) = |\mathcal{S}|, \quad (8)$$

where $\mathcal{S} = \mathcal{V}' \setminus R$, with R being the reference bus.

Remark 2: An observable subnetwork $\bar{G}\left(\mathcal{P}\right) = \left(\mathcal{V}', \mathcal{E}'\right)$ is closed in the sense that it consists of the buses $\mathcal{V}'$ and lines $\mathcal{E}'$ measured by $\mathcal{P}$. From (8), $\mathcal{P}$ contains at least a basic measurement set of $\mathcal{V}'$. Besides, $\bar{G}$ must include the reference bus R, i.e. $R \in \mathcal{V}'$, since otherwise $rank\left(\mathbf{H}_{\{\mathcal{P}\},\{\mathcal{S}\}}\right) < |\mathcal{S}|$.

We proceed to establish the equivalence between state observability and state estimate protection criterion.

Theorem 1: Protecting a set of meter measurements $\mathcal{P}$ can defend a set of state estimates $\mathcal{D}$ against undetectable attack, if and only if $\mathcal{D}$ is observable from $\mathcal{P}$.

Proof: We first prove the *if* part. When $\mathcal{D}$ is observable from $\mathcal{P}$, there must exist an observable subnetwork $\bar{G}\left(\bar{\mathcal{P}}\right) = \left(\bar{\mathcal{V}}, \bar{\mathcal{E}}\right)$ that includes $\mathcal{D}$, i.e. $\mathcal{D} \subseteq \bar{\mathcal{V}}$ and $\bar{\mathcal{P}} \subseteq \mathcal{P}$. From (8), we have $rank\left(\mathbf{H}_{\{\bar{\mathcal{P}}\},\{\bar{\mathcal{S}}\}}\right) = |\bar{\mathcal{S}}|$, where $\bar{\mathcal{S}} = \bar{\mathcal{V}} \setminus R$. Then,

the solution of $\mathbf{c}$ to $\mathbf{H}_{\{\bar{\mathcal{P}}\},*}\mathbf{c}=\mathbf{0}$ is $\mathbf{c}=\left(\mathbf{0},\mathbf{c}_{\mathcal{I}\setminus\bar{\mathcal{S}}}\right)^{\mathsf{T}}$, where $\mathbf{c}_{\mathcal{I}\setminus\bar{\mathcal{S}}}$ is an arbitrary vector. That is, no undetectable attack can be formulated to compromise $\bar{\mathcal{S}}$ if $\bar{\mathcal{P}}$ is well protected. Since $\mathcal{D}\subseteq\bar{\mathcal{S}}$ and $\bar{\mathcal{P}}\subseteq\mathcal{P}$, this completes the proof of the *if* part.

We then show the *only if* part. That is, there exists an undetectable attack to compromise $\mathcal{D}$ if $\mathcal{D}$ is unobservable from $\mathcal{P}$. Since random noise is not related to the network observability, we neglect $\mathbf{e}$ in (1). According to the definition of observability, there exists a $\mathbf{z}_{\mathcal{P}}$ and at least two different estimates of unknown variables, denoted by $\bar{\boldsymbol{\theta}}$ and $\dot{\boldsymbol{\theta}}$, satisfy

$$\mathbf{z}_{\mathcal{P}}=\mathbf{H}_{\{\mathcal{P}\},*}\bar{\boldsymbol{\theta}}=\mathbf{H}_{\{\mathcal{P}\},*},\dot{\boldsymbol{\theta}} \tag{9}$$

and $\bar{\theta}_k\neq\dot{\theta}_k$ for some $k\in\mathcal{D}$ when $\mathcal{D}$ is unobservable from $\mathcal{P}$. By letting $\mathbf{c}=\bar{\boldsymbol{\theta}}-\dot{\boldsymbol{\theta}}$, we have $\mathbf{H}_{\{\mathcal{P}\},*}\mathbf{c}=\mathbf{0}$ and $c_k\neq 0$. Therefore, an undetectable attack $\mathbf{a}=\mathbf{H}\mathbf{c}$ can compromise state θ_k without being detected. ■

Remark 3: All the unknown state estimates to be defended, i.e. $\mathcal{D}$, are included in an observable subnetwork constructed from a set of protected meters. In the following subsection, we find that the optimal observable subnetwork has an interesting Steiner tree structure.

B. Graphical equivalence of optimal protection

The power network observability analysis in [12] showed a connection between network observability and a spanning tree structure. The idea is briefly covered in Proposition 1.

Proposition 1: The measured full network $\bar{G}\left(\mathcal{M}\right)=\left(\mathcal{V},\mathcal{E}\right)$ is observable if and only if the graph defined on $\bar{G}$ contains a spanning tree, where each edge of which is mapped to a meter according to the following rules,

1) an edge is mapped to a flow meter placed on it, if any;
2) an edge without a flow meter is mapped to an injection meter that measures it;
3) different edges are mapped to different meters in $\mathcal{M}$.

Proof: See the proof in [12]. ■

Proposition 1 states that any basic measurement set of $\mathcal{V}$ can be mapped to a spanning tree in the measured full graph. On the other hand, a measured subnetwork $\bar{G}\left(\mathcal{P}\right)=\left(\bar{\mathcal{V}},\bar{\mathcal{E}}\right)$, where $\mathcal{P}\subseteq\mathcal{M}$, can also be considered as a closed network whose observability is only related to the components within $\bar{G}\left(\mathcal{P}\right)$. Therefore, there also exists a measurement-to-edge mapping in an observable subnetwork, specified as following.

Corollary 1: A measured subnetwork $\bar{G}\left(\mathcal{P}\right)=\left(\bar{\mathcal{V}},\bar{\mathcal{E}}\right)$ is observable if and only if the graph defined on $\bar{G}\left(\mathcal{P}\right)$ contains a tree that connects all vertices in $\bar{\mathcal{V}}$, where each edge of the tree is one-to-one mapped to a unique meter in $\mathcal{P}$ that takes its measurement.

Proof: The proof follows by replacing $\mathcal{M}$ with $\mathcal{P}$ in Proposition 1. ■

From Remark 3 and Corollary 1, we see that the unknown state estimates to be defended are indeed contained in a tree constructed from a protected meter measurement set. Therefore, we propose the following *minimum measured Steiner tree* (MMST) problem in a graph that is equivalent to the optimal state protection problem (7).

MMST problem: Given the measured full graph $\bar{G}\left(\mathcal{M}\right)=\left(\mathcal{V},\mathcal{E}\right)$. To protect a set of state estimates $\mathcal{D}$ with a minimum cost, the MMST problem finds a shortest Steiner tree $T^*=\left(\mathcal{V}^*,\mathcal{E}^*\right)$ (with the minimum number of edges) and a set of meters $\mathcal{P}^*\subseteq\mathcal{M}$ that satisfy the following conditions.

1) $\mathcal{V}^*$ is the set of all vertices measured by $\mathcal{P}^*$;
2) $\mathcal{D}\subset\mathcal{V}^*$ and $R\in\mathcal{V}^*$;
3) each edge in $\mathcal{E}^*$ is one-to-one mapped to a unique meter in $\mathcal{P}^*$ that takes its measurement.

Then, the set of meters $\mathcal{P}^*$ is the optimal solution to (7).

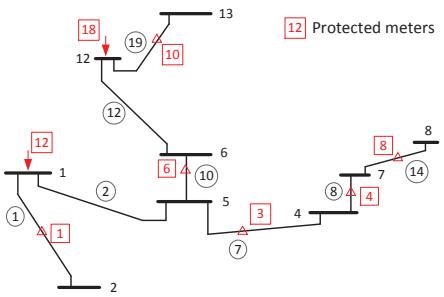

Fig. 2. An illustration of MMST from the IEEE 14-bus testcase.

We name the problem as a Steiner tree problem, instead of spanning tree, because T^* in general connects only a subset of vertices in the measured full graph. The three conditions ensure that all the unknown state estimates in T^*, including $\mathcal{D}$, are observable from $\mathcal{P}^*$. To bring out the intuitions, we present an example from Fig. 1, where we assume that $\mathcal{D}=\{v_8,v_{12}\}$ and v_1 is the reference bus. The optimal protected meters set $\mathcal{P}^*=\{r_1,r_3,r_4,r_6,r_8,r_{10},r_{12},r_{18}\}$ is obtained from exhaustive search. The corresponding minimum Steiner tree T^* is plotted in Fig. 2. We see that conditions 1) and 2) are clearly satisfied. Condition 3 is satisfied by mapping edges e_2 and e_{12} to injection meters r_{12} and r_{18}, and the other edges in $\mathcal{E}^*$ to the flow measurements placed on them.

We show that the MMST problem is *NP-hard* by considering a special case where flow meters are installed at all edges of $\bar{G}\left(\mathcal{M}\right)=\left(\mathcal{V},\mathcal{E}\right)$. Then, any Steiner trees that include R and $\mathcal{D}$ automatically satisfy the three conditions, i.e. by mapping each edge to the corresponding flow meter. In this case, the MMST problem becomes a standard minimum *Steiner tree* (MST) problem, which finds the shortest subtree of the full graph that connects R and all the vertices in $\mathcal{D}$. MST is a well-known *NP-hard* problem. The time complexity of known exact algorithms increase exponentially with $|\mathcal{D}|$ or $|\mathcal{I}|-|\mathcal{D}|$ [20]. Since MST is a special case of the MMST problem, the MMST problem is also *NP-hard* following the reduction lemma for computational complexity analysis. A special case of the MMST problem with $\mathcal{D}=\mathcal{I}$ is solved in [12] and [13] with time complexity $O\left(|\mathcal{V}||\mathcal{E}|\right)$. The special case is easy because $\mathcal{V}^*=\mathcal{V}$ holds automatically when all the state estimates are to be protected. The general MMST problem is much harder due to the combinatorial nature of possible $\mathcal{V}^*$.

IV. Graphical Methods for Optimal Protection

In this section, we first introduce two exact solution methods to solve the MMST problem, including the SVE method and a MILP formulation. Then, a tree pruning heuristic is proposed to obtain an approximate solution in polynomial time.

A. Steiner vertex enumeration algorithm

A vertex v in the Steiner tree solution $T^* = (\mathcal{V}^*, \mathcal{E}^*)$ is a *terminal* if $v \in \mathcal{D} \cup R$, or a *Steiner vertex* otherwise. The Steiner vertex enumeration (SVE) method enumerates the possible Steiner vertices $\mathcal{V}_0$ until a minimum observable subnetwork, including $\mathcal{V}_0$ and the terminals, is found. Then, $\mathcal{P}^*$ can be obtained by removing redundant measurements in the subnetwork using Gauss-Jordan elimination. A pseudo-code of the SVE is presented in Algorithm 1. The time complexity of SVE is $O\left(2^{|\mathcal{I}|-|\mathcal{D}|}\right)$, which is computational infeasible in large scale power networks, e.g. a 118-bus system. Therefore, we mainly use SVE as the performance benchmark to evaluate the algorithms proposed in the following subsections.

Algorithm 1: Steiner vertex enumeration algorithm

input : $\mathcal{I}, \mathcal{D}, \mathcal{M}, R$
output: Minimum protected measurements $\mathcal{P}^*$ to defend $\mathcal{D}$
repeat
  Enumerate a set of Steiner vertices $\mathcal{V}_0 \subseteq \{\mathcal{I} \setminus \mathcal{D}\}$, from size $|\mathcal{V}_0| = 0$ to $|\mathcal{I}| - |\mathcal{D}|$. Let $\bar{\mathcal{S}} = \mathcal{D} \cup \mathcal{V}_0$;
  Find the meters $\bar{\mathcal{P}}$ that measure only the buses in $\bar{\mathcal{S}} \cup R$;
until $rank\left(\mathbf{H}_{\{\bar{\mathcal{P}}\},\{\bar{\mathcal{S}}\}}\right) = |\bar{\mathcal{S}}|$;
$\mathcal{P}^* =$ a basic measurement set of $\bar{\mathcal{S}}$;

B. Mixed integer linear programming formulation

In this subsection, we propose a MILP formulation to solve the MMST problem, which has much lower complexity than SVE by exploiting the optimal solution structure. Consider a digraph $\overrightarrow{G} = (\mathcal{V}, \mathcal{A})$ constructed by replacing each edge in the measured full graph $\bar{G}(\mathcal{M}) = (\mathcal{V}, \mathcal{E})$ with two arcs in opposite directions. We set the reference bus as the root and allocate one unit of demand to each vertex in $\mathcal{D}$. Commodities are sent from the root to the vertices in $\mathcal{D}$ through some arcs. Then, the vertices in $\mathcal{D}$ are connected to R via the used arcs if and only if all the demand is satisfied. When we require using the minimum number of arcs to deliver the commodity, the used arcs will form a directed tree, referred to as a *Steiner arborescence*. Evidently, the solution to the MMST problem can be obtained if we solve the following *minimum measured Steiner arborescence* (MMSA) problem and neglect the orientations of the arcs. Without causing confusions, we say an arc (i, j) is measured by a meter if the edge $[i, j]$ in $\bar{G}(\mathcal{M})$ is measured by the meter.

MMSA problem: Given a digraph $\overrightarrow{G} = (\mathcal{V}, \mathcal{A})$, find the shortest arborescence $\overrightarrow{T}^* = (\mathcal{V}^*, \mathcal{A}^*)$ and a set of meters $\mathcal{P}^* \subseteq \mathcal{M}$ that satisfy the following conditions

1) $\mathcal{V}^*$ is the set of all vertices measured by $\mathcal{P}^*$;
2) $\mathcal{D} \subset \mathcal{V}^*$ and $R \in \mathcal{V}^*$;
3) each arc in $\mathcal{A}^*$ is one-to-one mapped to a unique meter in $\mathcal{P}^*$ that takes its measurement.

From condition 1), if an arc in $\overrightarrow{T}^*$ is mapped to an injection meter, all the vertices measured by the injection meter must also be included in the arborescence like the terminals, as if an extra demand is allocated at these vertices. To distinguish from the actual demand at $\mathcal{D}$, we refer to the extra demand induced by the use of injection meters as *pseudo demand*. Then, the MMSA problem is to satisfy both the actual and pseudo demand using minimum number of arcs.

For an arc $(i, j) \in \mathcal{A}$, let x_{ij} be a binary variable with $x_{ij} = 1$ indicating that the arc is included in $\overrightarrow{T}^*$ and 0 otherwise. y_{ij} denotes the total amount of commodity through (i, j). z_{ij} be a binary variable with $z_{ij} = 1$ indicating that the injection meter at vertex i is mapped to arc (i, j) or (j, i), and 0 otherwise. Then, a MILP formulation of the MMSA problem is

$$\min_{\mathbf{X},\mathbf{Y},\mathbf{Z}} \quad \sum_{(i,j)\in\mathcal{A}} x_{ij} + \frac{1}{w}\sum_{(i,j)\in\mathcal{A}} z_{ij} \tag{10a}$$

$$\text{s. t.} \quad x_{ij} \geq \frac{y_{ij}}{w}, \ \ \forall (i,j) \in \mathcal{A} \tag{10b}$$

$$\mathbf{1}_E(i,j) + z_{ij} + z_{ji} \geq x_{ij}, \ \forall (i,j) \in \mathcal{A} \tag{10c}$$

$$\sum_{(i,j)\in\mathcal{A}} z_{ij} \leq \mathbf{1}_V(i), \ \ \forall i \in \mathcal{V} \tag{10d}$$

$$\sum_{(i,j)\in\mathcal{A}} y_{ij} - \sum_{(j,k)\in\mathcal{A}} y_{jk} = d(j), \forall j \in \mathcal{V} \setminus R \tag{10e}$$

$$x_{ij}, z_{ij} \in \{0,1\}, \ y_{ij} \geq 0, \forall (i,j) \in \mathcal{A}. \tag{10f}$$

Here, w is chosen as a large positive number such that $w > \sum_{(i,j)\in A} z_{ij}$ and $w > y_{ij}$ always hold. $\mathbf{1}_E(i,j)$ and $\mathbf{1}_V(i)$ are two binary indicator functions, where $\mathbf{1}_E(i,j) = 1$ if a flow meter is available at edge $[i, j]$ and $\mathbf{1}_V(i) = 1$ if an injection meter is available at v_i. $d(j)$ is the demand at vertex j, where

$$d(j) = \begin{cases} 1 + \sum_{(j,k)\in\mathcal{A}} z_{jk} + \sum_{[k,j]\in\mathcal{E}} \sum_{(k,s)\in\mathcal{A}} z_{ks} & j \in \mathcal{D} \\ \sum_{(j,k)\in\mathcal{A}} z_{jk} + \sum_{[k,j]\in\mathcal{E}} \sum_{(k,s)\in\mathcal{A}} z_{ks} & j \notin \mathcal{D}. \end{cases}$$

For $j \notin \mathcal{D}$, $d(j)$ is the total pseudo demand. Otherwise, one extra unit of actual demand is counted as well.

As we can see, there are two terms in (10a), each corresponding to one objective. The first term is to minimize the total number of arcs included in the arborescence. The second term is is to minimize the number of injection measurements. Notice that the first objective is primary, as the second term in (10a) is always dominated by the first one due to the scaling factor $1/w$, which makes the second term always less than 1. As such, (10a) is to minimize the total number of arcs in the arborescence, and meanwhile eliminating redundant injection measurements, such as the case when two injection measurements are assigned to the same arc. Constraint $(10b)$ forces arc (i, j) to be included in $\overrightarrow{T}^*$ if any commodity flow passes through (i, j). Constraint (10c) and (10d) ensure that each arc (i, j) included in $\overrightarrow{T}^*$ has at least one measurement assigned to it and each injection measurement can only be assigned to at most one arc. The flow conservative constraint (10e), together with $(10b)$, forces the selected arcs to form an arborescence rooted at the reference vertex and spanning all vertices with positive demand. Once the optimal solution to (10) is obtained, we can restore the optimal solution $\mathcal{P}^*$ to the MMST problem by including: 1) injection measurement on bus i if $z_{ij} = 1$, $\forall (i, j) \in \mathcal{A}$; 2) flow measurement on arc (i, j), if $x_{ij} = 1$ and $z_{ij} = z_{ji} = 0$, $\forall (i, j) \in \mathcal{A}$. That is, the arcs in $\overrightarrow{T}^*$ not mapped to any injection measurement.

Extensive simulations show that the MILP formulation always obtains the same optimal solution as the SVE algorithm. The detailed experiment setup is omitted due to the page limit. The MILP significantly reduces the computational complexity

by exploiting the solution structure. For instance, a problem in a 57-bus system that is computationally infeasible by the SVE algorithm can now be solved by the MILP within minutes. Nonetheless, the computational complexity of the state-of-art MILP algorithms, such as branch and bound and cutting-plane method, etc, still grows exponentially with the problem size. We observe from simulations that it takes excessively long time to solve the problem in a 300-bus power system.

C. Tree pruning heuristic

To tackle the intractability of the problem, we propose a tree-pruning based heuristic (TPH) that finds an approximate solution in polynomial time. We refer to a tree $T = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$, along with a set of measurement $\bar{\mathcal{P}}$, a *feasible measured tree* if T and $\bar{\mathcal{P}}$ satisfy the conditions of the MMST problem. Our observation is that, although it is hard to find a MMST, it is relatively "easy" to find a feasible tree that includes all the vertices in the graph using the techniques in [12]. Starting from a feasible measured tree that spans all vertices in the measured full graph, our TPH method iteratively prunes away redundant vertices and updates the feasible tree, until a shortest possible tree is obtained. A pseudo-code is provided in Algorithm 2. The TPH consists of multiple rounds of pruning operations. Here, we explain one round of pruning, which corresponds to line 2-8 in the pseudo-code, in the following 4 steps.

Algorithm 2: Tree pruning heuristic algorithm

input : $\bar{G}(\mathcal{M}) = (\mathcal{V}, \mathcal{E})$, $\mathcal{D}$, R, K
output: Minimum protected measurements $\mathcal{P}^*$ to defend $\mathcal{D}$
1 **initialization:** $\bar{\mathcal{V}} = \mathcal{V}$;
2 **repeat**
3 Let $W = |\bar{\mathcal{V}}|$. Find K basic measurement sets of $\bar{\mathcal{V}}$, denoted by $\bar{\mathcal{P}}^k$, $k = 1, .., K$. For each $\bar{\mathcal{P}}^k$, construct a feasible measured trees T_k;
4 **for** each T_k **do**
5 Starting from R to all leaf vertices, find the largest prunable subset $C_s^*(i)$ for each v_i. Update $T_k = T_k \setminus \{C_s^*(i) \cup D(C_s^*(i))\}$ until each vertex in T_k is either processed or pruned;
6 **end**
7 Select the minimum trees T^* and update $\bar{\mathcal{V}}$ by letting $\bar{\mathcal{V}} =$ the vertices in T^*;
8 **until** $W = |T^*|$;
9 $\mathcal{P}^* =$ the remaining measurements corresponding to T^*;

Step 1: Feasible tree generation. For a set of vertices $\bar{\mathcal{V}}$ (initially set to be $\mathcal{V}$), we generate K feasible measured trees, where K is a tunable parameter (lines 3-4). In this step, we first find the meters that measure only the vertices in $\bar{\mathcal{V}}$. Among them, we find K basic measurement sets of $\bar{\mathcal{V}} \setminus R$, denoted by $\bar{\mathcal{P}}^k$ ($k = 1, .., K$), using Gauss-Jordan elimination. Then, we construct K feasible spanning trees $T_k = (\bar{\mathcal{V}}, \bar{\mathcal{E}}^k)$, one for each $\bar{\mathcal{P}}^k$, using the max-flow method given in the Appendix.

Step 2: Vertex identification. For each tree T_k, we identify the child and descendant vertices of each vertex (included in line 5-6 in Algorithm 2). This can be achieved by constructing a directed tree from the root to all leaf vertices. If there is an arc (i, j), we say v_j is a child of v_i, denoted by $v_j \in C(i)$. In general, if there exists a path from v_i to v_j, we refer to v_j as a descendent of v_i, denoted by $v_j \in D(i)$. In Fig. 3, for instance, v_6 and v_7 are the child vertices of v_4, while v_6 to v_{13} are all descendent vertices of v_4.

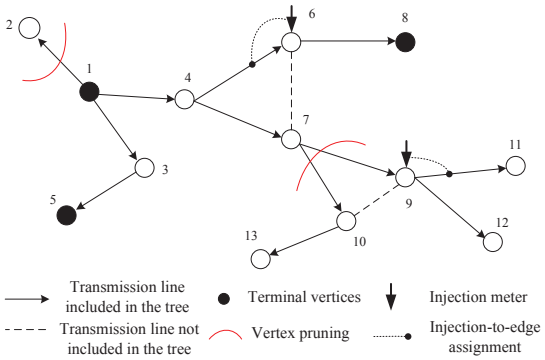

Fig. 3. A measured feasible tree. $\{v_1, v_5, v_8\}$ are the terminals and v_1 is the reference. Two marked edges ($[4, 6]$ and $[9, 11]$) are mapped to injection meters and the other unmarked edges are mapped to flow meters.

Step 3: Tree pruning. For each T_k, we start from the root to the leaf vertices to prune away redundant vertices (line 5-6). For a vertex v_i, we find the largest prunable subset $C_s^*(i) \subseteq C(i)$, such that the residual tree is still a feasible measured tree after all the vertices in $\{C_s(i) \cup D(C_s(i))\}$ are pruned. In particular, $\{C_s(i) \cup D(C_s(i))\}$ can be pruned if:

1) $\{C_s(i) \cup D(C_s(i))\}$ contains no terminal vertex,
2) the deletion of $\{C_s(i) \cup D(C_s(i))\}$ will remove all the edges mapped to injections that measure any vertex in $\{C_s(i) \cup D(C_s(i))\}$.

Then, we update T_k by removing all the vertices in $\{C_s^*(i) \cup D(C_s^*(i))\}$ and proceed to another vertex until each vertex in $\bar{\mathcal{V}}$ is either checked or pruned.

Step 4: Vertex update. Let $|T_k|$ be the number of remaining vertices in T_k. Then, we select among the K trees the one with minimum vertices, denoted by T^*. If $|T^*| = |\bar{\mathcal{V}}|$, i.e. no vertex is removed for all the K trees, we terminate the algorithm and output $\mathcal{P}^*$ as the remaining meters in T^* (line 7-9). Otherwise, we update $\bar{\mathcal{V}}$ as the remaining vertices in T^* and start another round of pruning from Step 1) .

In Fig. 3, we present an example to illustrate TPH. Starting from the root v_1, among the three child vertices of v_1, only v_2 can be pruned, since the descendent vertices of either v_3 or v_4 contain terminal vertex. After pruning v_2, we proceed to check v_3, whose only child vertex is a terminal. Then, we check v_4, where neither of its child vertices v_6 and v_7 can be pruned separately or together. This is because v_6 contains terminal as its descendent vertices, and the removal of v_7 does not remove the edge $[4, 6]$, which is mapped to the injection meter at v_6 that measures v_7. For v_7, however, all of its descendent vertices can be pruned following the vertices pruning conditions. Up to now, we have finished the first round of pruning. Then, we use the remaining vertices $\{v_1, v_3, v_4, v_5, v_6, v_7, v_8\}$ to generate new feasible trees, if any, and repeat the pruning operations iteratively until no vertex can be further pruned.

The purpose of introducing the parameter K is because the final output $\mathcal{P}^*$ is closely related to the tree's topology obtained in Step 1. Intuitively, with larger K, we have a larger chance to obtain a smaller $|\mathcal{P}^*|$ but also consume more computations. The proper choice of K will be discussed in Simulations. The correctness of TPH is obvious from the

TABLE I
STATISTICS OF DIFFERENT POWER SYSTEM TESTCASES

No. of buses	14-bus	57-bus	118-bus
No. of lines	20	80	186
Total no. of measurements	19	80	180
No. of inject measurements	8	30	70
No. of flow measurements	11	50	110
No. of unmeasured lines	2	2	7

following facts: 1) the K residual trees are always feasible measured tree; 2) the size of the minimum residual tree is non-increasing during the iterations; 3) $|\mathcal{P}^*|$ equals the size of the minimum residual tree. There are at most $|\mathcal{I}| - |\mathcal{D}|$ rounds of pruning. In each round, K trees are pruned and each takes $O\left(|\mathcal{I}|^3\right)$ time complexity, dominated by the Gauss-Jordan elimination computation. The overall time complexity is $O\left(K|\mathcal{I}|^4\right)$, which is considered efficient even for very large scale power systems.

V. SIMULATION RESULTS

In this section, we use simulations to evaluate the proposed defending mechanisms. All the computations are solved in MATLAB on a computer with an Intel Core2 Duo 3.00-GHz CPU and 4 GB of memory. In particular, MatlabBGL package is used to solve some of the graphical problems [21], such as maximum-flow calculation, etc. Besides, Gurobi is used to solve MILP problems [22]. The power systems we considered are IEEE 14-bus, 57-bus and 118-bus testcases, whose topologies are obtained from MATPOWER [23] and summarized in Table I. All the systems are observable with the respective measurement placement. For illustration purpose, a measurements placement of the 14-bus system is plotted in Fig. 1. The measurement placements for 57-bus and 118-bus systems are omitted for the simplicity of expositions.

We first evaluate the computational complexity of TPH in Fig. 4, where MILP is the benchmark for comparison. For TPH, we set the parameter $K = 1$ and record the total number of vertices that are checked to produce a solution. For MILP, we record the number of nodes explored in the search tree by the branch-and-bound algorithm. Both numbers are the iterations consumed by the two methods to obtain a solution. Besides, we also record the CPU time for both methods. The results in Fig. 4 are the average performance of 50 independent experiments. Without loss of generality, we randomly generate a $\mathcal{D}$ with size $|\mathcal{D}| = 4$ in each experiment. In Fig. 4a, we show the average number of iterations for 14-bus, 57-bus and 118-bus systems, respectively. We find that the iteration numbers are very close for both methods in the 14-bus system, where TPH consumes 38 iterations and the MILP consumes 47 iterations to obtain a solution. However, the difference becomes more and more significant as the network size increases. The number of iterations of TPH increases by 11 times as the network size increases from 14 to 118 buses. In vivid contrast, the iteration number of MILP increases rapidly by 2272 times, from merely 47 to 106787. Similar results are also observed for the CPU time, where TPH takes only 0.485 second to obtain a solution in 118-bus system, while MILP consumes around 5 minutes, which is 1410 times slower than in the 14-bus system. The booming computational complexity

TABLE II
PERFORMANCE OF TPH AND MILP IN 57-BUS TESTCASE

$\lvert\mathcal{D}\rvert$	1	4	9	19	29	39	49
$\lvert\mathcal{P}\rvert$, $K=1$	11.8	22.2	30.3	39.5	46.3	51.6	55.8
$\lvert\mathcal{P}\rvert$, $K=3$	10.7	20.8	28.0	37.0	43.0	48.8	54.1
$\lvert\mathcal{P}\rvert$, $K=5$	9.9	20.4	27.8	36.7	42.5	47.9	53.7
$\lvert\mathcal{P}\rvert$, $K=10$	9.7	20.2	27.3	36.3	42.1	47.6	53.1
$\lvert\mathcal{P}\rvert$, $K=15$	9.4	20.0	26.8	35.9	41.7	47.3	52.8
MILP ($\lvert\mathcal{P}^*\rvert$)	8.8	18.2	25.4	34.6	40.7	46.2	51.8
Gap	0.6	1.8	1.4	1.3	1.0	1.1	1.0

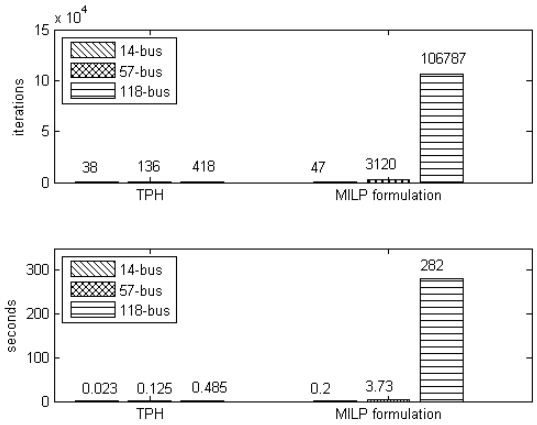

Fig. 4. Comparison of computational complexity for MILP and TPH. (a) The figure above shows the average number of iterations to obtain a solution; (b) the figure below shows the average CPU time to obtain a solution.

of the MILP method is due to the NP-harness of solving a MILP. It is foreseeable that the computational complexity of the MILP method will become extremely expensive as we further increase the network size. For instance, the projected CPU time of MILP to solve a problem in 300-bus system is more than 5 days, while it takes TPH less than 2 seconds.

We also investigate the impact of the parameter K to the performance of TPH. By varying the values of K and $|\mathcal{D}|$, we show in Table III the average solution size $|\mathcal{P}|$ of TPH and MILP. Each entry of the table is the average performance of 50 independent experiments. From the 2nd to the 6th rows, we see that better solution, i.e. smaller $|\mathcal{P}|$, is obtained with larger K. Compared with the optimal solution $\mathcal{P}^*$ obtained by MILP, TPH protects on average only 1.13 more meters when $K = 15$. The optimality gap is less than 10% for all the cases. For better visualization, we plot the ratio $|\mathcal{P}|/|\mathcal{P}^*|$ for some selected $|\mathcal{D}|$'s in Fig. 5a. We notice that the ratio improves notably for small $|\mathcal{D}|$ as K increases from 1 to 15. For instance, the ratio improves from 1.32 to 1.04 for $|\mathcal{D}| = 1$. The improvement is especially notable when we change $K = 1$ to 3. However, the improvement becomes marginal as we further increase K, such as the case with $|\mathcal{D}| = 49$, where the ratio only improves by 0.03 from $K = 1$ to 15. We also plot in Fig. 5b the CPU time normalized against the time consumed when $K = 1$. We observe that the CPU time increases almost linearly with K, which matches our analysis in Section IV. Results in Fig. 5 indicate that we should select a proper K to achieve a balance between the quality of approximate solution and computational complexity. In particular, a large K, such as $K = 10$, should be used when $|\mathcal{D}|$ is small relative to n, while small K, such

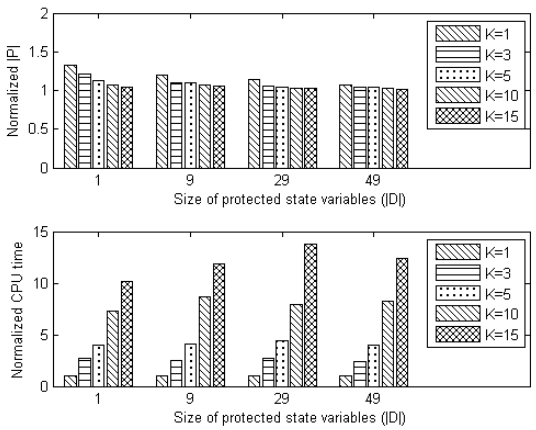

Fig. 5. Effect of K to the performance of TPH in the 57-bus system. (a) The figure above shows the solution size of TPH normalized by the optimal solution size obtained by MILP; (b) the figure below shows the CPU time of TPH normalized by the CPU time when $K = 1$.

as $K = 3$, should be used when $|\mathcal{D}|$ is relatively large.

VI. Conclusions

In this paper, we used graphical methods to study defending mechanisms that protect a set of state estimates from false-data injection attacks. By characterizing the optimal protection problem into a variant Steiner tree problem, we proposed both exact and approximate algorithms to select the minimum number measurements for system protection. The advantageous performance of the proposed defending mechanisms has been evaluated in IEEE standard power system testcases.

Appendix
Maximum-flow method for tree construction

We use an example in Fig. 1 to illustrate the method to obtain a feasible spanning tree. We consider a basic measurement set $\bar{\mathcal{P}} = \{r_1, r_6, r_{12}, r_{14}\}$ of $\bar{\mathcal{V}} \setminus R$, where $\bar{\mathcal{V}} = \{v_1, v_2, v_4, v_5, v_6\}$ and $R = v_1$. The set of edges measured by $\bar{\mathcal{P}}$ is $\bar{\mathcal{E}} = \{e_1, e_2, e_5, e_7, e_{10}\}$. Then, a directed graph is constructed in Fig. 6, where v_1 is chosen as the root to construct the spanning tree. We select in advance an edge connected to the root, say e_1, in the final tree solution. This is achieved by setting both the lower and upper capacity bounds of the edge to be 1. The other edges' lower and upper capacity bounds are set to be 0 and 1, respectively. Then, a maximum flow is calculated from s to t. If the problem is feasible, i.e. the flow solution is 1 in edge e_1, we obtain a measurement-to-edge mapping by observing the saturating flows in the graph. Otherwise, we select another edge connected to the root and recalculate the maximum flow problem. Since $\bar{\mathcal{V}}$ is observable from $\bar{\mathcal{P}}$, there is always a solution. In the above example, the final measurement-to-edge mapping is $\{r_1, r_6, r_{12}, r_{14}\} \leftrightarrow \{e_1, e_{10}, e_2, e_7\}$. Then, the edges obtained by the maximum flow calculation will form a tree that spans all vertices in $\bar{V}$.

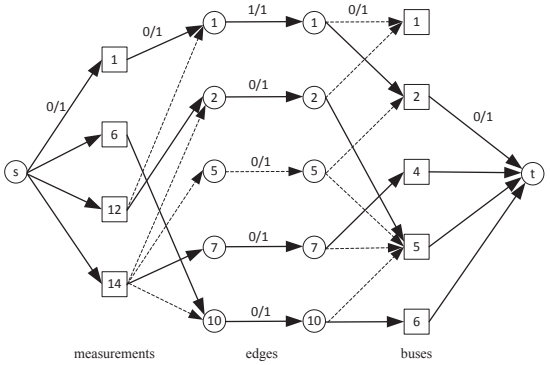

Fig. 6. Maximum-flow method for measurement tree construction. The solid arcs are the saturated arcs obtained in the final solution.

References

[1] A. Abur and A. Gomez Exposito, "Power System State Estimation: Theory and Implementation". New York: Marcel Dekker, 2004.

[2] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the* 16*th ACM conference on Computer and communications security*, Chicago, Illinois, 2009, pp. 21-32.

[3] A. Teixeira, G. Dan, H. Sandberg, and K. H. Johansson, "Cyber security study of a scada energy management system: stealthy deception attacks on the state estimator," in *IFAC World Congress*, Milan, Italy, 2011.

[4] L. Jia, R. J. Thomas and L. Tong, "Impacts of Malicious Data on Real-Time Price of Electricity Market Operations," *45th HICSS*, pp.1907-1914, Jan 4-7, 2012.

[5] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Trans. on smart grid*, vol.2 (4), pp.659-665, Dec 2011.

[6] Y. Yuan, Z. Li, K. Ren, "Modeling Load Redistribution Attacks in Power Systems," *IEEE Trans. on smart grid*, vol.2 (2), pp.382-390, June 2011.

[7] R. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye, "Detecting false data injection attacks on DC state estimation," in *CPSWEEK 2010.*

[8] O. Vukovic, K. .C. Sou, G. Dan and H. Sandberg, "Network-aware mitigation of data integrity attack on power system state estimation", *IEEE JSAC*, vol.30, no.6, July 2012.

[9] T. T. Kim and H. V. Poor, "Stategic protection against data injection attack on power grids", *IEEE Trans. on smart grid*, vol.2(2), June 2011.

[10] B. Milosevic and M. Begovic, "Voltage-Stability Protection and Control Using a Wide-Area Network of Phasor Measurements," *IEEE trans actions on power systems*, VOL. 18, NO.1, Feb 2003.

[11] A. Von. Meier. *Electric power systems: a conceptual introduction.* Wiley-IEEE Press, 2006.

[12] G. R. Krumpholz, K. A. Clements and P. W. Davis, "Power System Observability: A Practical Algorithm Using Network Topology," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-99, no.4, pp. 1534-1542, July 1980.

[13] A. Barglela, M. R. Irving, M. J. H. Sterling, "Observability determination in power system state estimation using a network flow technique", *IEEE Transactions on power systems*, Vol. PWRS-1, No.2, May 1986.

[14] O. Kosut, L. Jia, R. J. .Thomas and L. Tong, "Malicious data attack on the smart grid", *IEEE trans. on smart grid*, vol.2(4), pp. 645-658, 2011.

[15] K. C. Sou, H. Sandberg and K. H. Johansson, "Electricity power network security analysis via minimum cut relaxation", *50th CDC-ECC*, pp. 4054-4059, Dec. 2011.

[16] H. Sandberg, A. Teixeira and K. H. Johansson, "On security indices for state estimators in power networks," in *CPSWEEK 2010.*

[17] K. C. Sou, H. Sandberg and K. H. Johansson, "Computing critical k-tuples in power networks", *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1511-1520, 2012.

[18] S. Bi and Y. J. Zhang, "Defending mechanisms against false-data injection attacks in the power system state estimation", *In Proc. of IEEE Globecom SG-COMNETS*, Houston, TX, Dec. 2011.

[19] J. G. Grainger, W. D. Stevenson. Jr., "Power System Analysis", McGraw-Hill, 1994.

[20] F. K. Hwang, D. S. Richards and P. Winter, "The Steiner tree problem", *Monograph in Annals of Discrete Mathematics*, 53, Elsevier, 1992.

[21] D. Gleich, Contents Matlab BGL v4.0, 2006. [Online]. Available: http://www.stanford.edu/ dgleich/programs/matlab_bgl/.

[22] Gurobi, [Online]. Available: http://www.gurobi.com/.

[23] R. D. Zimmerman and C. E. Murillo-Sanchez, "MATPOWER, A MATLAB power system simulation package." [Online] Available: http://www.pserc.cornell.edu/matpower/manual.pdf, September 2007.